Statistics Surveys
Vol. 0 (2008) 1–24
ISSN: 1935-7516

An Overview of Mixture Models

Derek S. Young

Department of Statistics
The Pennsylvania State University
University Park, Pennsylvania 16802-1294
e-mail: dsy109@stat.psu.edu

Abstract: With the advancement of statistical theory and computing power, data sets are providing a greater amount of insight into the problems of today. Statisticians have an ever increasing number of tools to attack these problems, some of which can be implemented in the area of mixture modeling. There is a great deal of literature on mixture models and this work attempts to provide a general overview of the subject, including the discussion of relevant issues and algorithms. The reader can hope to gain a broad understanding of concepts in mixture modeling and find the references cited within as a valuable resource for the next stage of their research.

Keywords and phrases: Bayesian methods, bootstrapping, identifiability, EM algorithm, label switching.

Contents

1. Introduction

Finite mixture models have long been used as a way to model a sample of observations that arise from a number of (usually) *a priori* known classes with

unknown proportions. For example, consider inferring the strategies young children employ when presented with a cognitive task. The children's performance on the task may be modeled using finite mixtures with the components pertaining to the different strategies (Thomas and Horton (1997)). Or perhaps, consider attempting to classify a group of individuals by the way they each speak the same word - a challenging problem due to context dependencies (i.e. different points in time, gender, etc.) in speech recognition. Here, a finite mixture may be used with the components pertaining to different 'vowel classes' of spoken words (Peng et al. (1996)). Or finally, consider assessing the service quality of banking institutions. The institutions may be modeled using finite mixtures with components pertaining to different market segments (Wedel and DeSarbo (1994)).

Studies such as those mentioned above have the roots of their analyses grounded in a seminal work by Pearson (1894). In this article, Pearson (1894) was one of the first individuals to incorporate the use of mixture models as well as note some of the issues surrounding them - in particular estimation and identifiability. These are issues still prominent in today's mixture research and they will be addressed in this work.

Since the time of the Pearson (1894) article, a great deal of literature has emerged in many disciplines regarding mixture models. In addition to the numerous technical and cross-disciplinary articles on mixture modeling, monographs concerning the subject include Everitt and Hand (1981), Titterington et al. (1985), Lindsay (1995), and McLachlan and Peel (2000). This article addresses many of the issues presented in these monographs, as well as current work, but at a high-level overview.

2. A General Mixture Model

Suppose we have n subjects where we take a series of m measurements, say $\mathbf{Y}_i = (Y_{i,1}, \ldots, Y_{i,m})^{\mathrm{T}}$, on the i^{th} subject for $i = 1, \ldots, n$. Furthermore, take $\mathbf{y}_1, \ldots, \mathbf{y}_n$ as realized values of the $\mathbf{Y}_i$'s, which are independent and identically distributed (*iid*) according to a distribution F. In this scenario, standard multivariate techniques (Johnson and Wichern (2002) and Anderson (2003)) can be employed to estimate the common population mean vector, $\boldsymbol{\mu}$, and the population variance-covariance matrix, Σ.

Suppose, in addition to the above scenario, there is an assumed heterogeneity with respect to the response tendencies of the subjects. One way to account for this heterogeneity is by suggesting k different classes to which the subjects could essentially belong. Assuming fixed k, the distribution of the $\mathbf{Y}_i$'s has the *k-component mixture density*

$$f_k(\mathbf{y}_i; \boldsymbol{\psi}) = \sum_{j=1}^{k} \lambda_j g_j(\mathbf{y}_i; \boldsymbol{\theta}_j), \tag{2.1}$$

where $\lambda_j > 0$ and $\sum_{j=1}^{k} \lambda_j = 1$ are the weights (or *mixing proportions*) for the components of the model. (The subscript for f will be suppressed except when

to stress the dependence of f on k.) Furthermore, define

$$\Lambda_{k-1} = \left\{ (\lambda_1, \lambda_2, \ldots, \lambda_{k-1}) : \sum_{j=1}^{k-1} \lambda_j < 1,\ \lambda_j \in (0,1)\ \forall j \right\} \subset \mathbb{R}^{k-1},$$

where λ_k has been arbitrarily omitted since $\lambda_k = 1 - \sum_{j=1}^{k-1} \lambda_j$. The g_j's are known component densities, parameterized by $\boldsymbol{\theta}_j \in \Theta_j \subseteq \mathbb{R}^{q_j}$ such that Θ_j represents the specified parameter space for the $\boldsymbol{\theta}_j$'s. The mixture density f is parameterized by $\boldsymbol{\psi} \in \boldsymbol{\Psi}$ such that $\boldsymbol{\Psi}$ represents the specified parameter space for all unknown parameters in the mixture model. Note that

$$\boldsymbol{\Psi} = \left(\prod_{j=1}^{k} \Theta_j \right) \times \Lambda_{k-1},$$

where $\boldsymbol{\Psi} \subset \mathbb{R}^r$ and $r = (\sum_{j=1}^{k} q_j) + k - 1$. We will take F as the corresponding k-component mixture distribution whose components are composed of the distributions G_j. For the scenarios presented in this work, the G_j differ only in $\boldsymbol{\theta}_j$, thus we will take $g_j \equiv g$ and $q_j \equiv q$ which yields $\boldsymbol{\Psi} = \Theta^k \times \Lambda_{k-1}$ and $r = kq + k - 1$.

3. Estimation

In this section, we focus on estimation of the parameters of a mixture model, $\boldsymbol{\psi}$, given $\mathbf{Y}$ and k. Early works employed a method of moments approach (for instance, Pearson (1894) and Quandt and Ramsey (1978)), but with the advent of more efficient computing, numerous algorithms have emerged as tools for estimation in the mixture setting. These techniques can be classified into two primary categories: likelihood methods and Bayesian methods. We will provide a brief literature review on some of the available techniques and provide a more complete description of a few commonly employed algorithms. We will close this section with one final issue which concerns estimating the number of components when this is not known *a priori*.

3.1. Likelihood Methods

The likelihood for the parameters of a mixture model, $\boldsymbol{\psi}$, can be easily formulated using the mixture density in (2.1) as

$$L(\boldsymbol{\psi}; \mathbf{y}) = \prod_{i=1}^{n} f(\mathbf{y}_i; \boldsymbol{\psi}).$$

In dealing with likelihood methods, it is often easier to work with the log likelihood:

$$\ell(\boldsymbol{\psi}) = \log L(\boldsymbol{\psi}; \mathbf{y}) = \sum_{i=1}^{n} \log f(\mathbf{y}_i; \boldsymbol{\psi}). \tag{3.1}$$

Then, an estimate $\hat{\boldsymbol{\psi}}$ (the MLE) is provided by solving

$$S(\mathbf{y};\boldsymbol{\psi}) = \frac{\partial \ell(\boldsymbol{\psi})}{\partial \boldsymbol{\psi}} = \mathbf{0}, \tag{3.2}$$

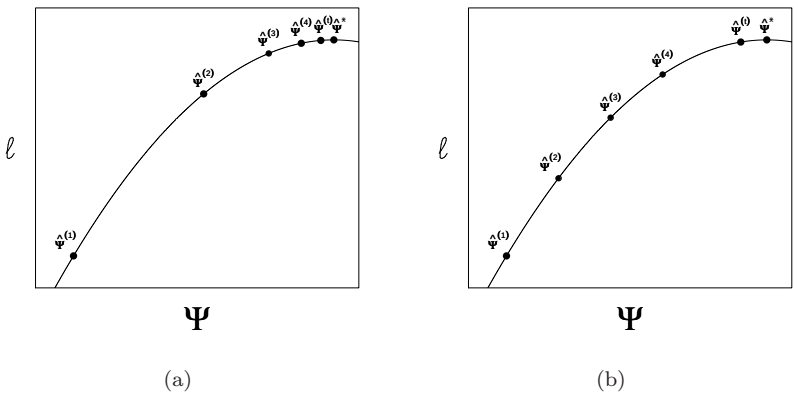
where $S(\mathbf{y};\boldsymbol{\psi})$ is called the *score function*.

It is necessary to consider the possibility of multiple local maxima since the likelihood will have multiple roots. Moreover, the likelihood function may be unbounded, which becomes a considerable concern when implementing various algorithms (as will be discussed). Focusing on local maxima on the interior of the parameter space (denoted by $\boldsymbol{\Psi}^{\circ}$) helps circumvent this problem because under certain regularity conditions, there exists a strongly consistent sequence of roots to the likelihood equation that is asymptotically efficient (see Ferguson (1996)). In fact, a $\sqrt{n}$-consistent estimator can be constructed using the method of moments estimator mentioned earlier. For a deeper treatment of the choice of root, as well as testing for a consistent root, refer to McLachlan and Peel (2000).

The rate of convergence for the likelihood methods to be discussed will also be considered. Consider a norm $\|\cdot\|$ on $\boldsymbol{\Psi}$ and a sequence c_t such that $c_t \to 0$ as $t \to \infty$. If the sequence of iterates $\{\boldsymbol{\psi}^{(t)}\}$ draws sufficiently close to a solution $\boldsymbol{\psi}^*$ of (3.2), then the *rate of convergence* is given by

$$\|\boldsymbol{\psi}^{(t+1)} - \boldsymbol{\psi}^*\| \leq c_t \|\boldsymbol{\psi}^{(t)} - \boldsymbol{\psi}^*\|^q, \tag{3.3}$$

where $t = 0, 1, \ldots$ and $q \geq 1$. When replacing the sequence $\{c_t\}$ by a constant c, then we refer to q in (3.3) as a *local rate of convergence*. An illustration of local linear ($q = 1$) and local quadratic ($q = 2$) convergence is given in Figure 1.



(a) (b)

FIG 1. *Illustration of (a) local quadratic convergence and (b) local linear convergence.*

3.1.1. Newton Methods

An efficient way for solving (3.2) is to implement a Newton-type method. The Newton-Raphson method takes a linear Taylor series expansion about the current fit $\boldsymbol{\psi}^{(t)}$ for $\boldsymbol{\psi}$, which yields

$$S(\mathbf{y};\boldsymbol{\psi}) \approx S(\mathbf{y};\boldsymbol{\psi}^{(t)}) - I(\boldsymbol{\psi}^{(t)};\mathbf{y})(\boldsymbol{\psi}-\boldsymbol{\psi}^{(t)}), \tag{3.4}$$

where

$$I(\boldsymbol{\psi};\mathbf{y}) = -\frac{\partial S(\mathbf{y};\boldsymbol{\psi})}{\partial \boldsymbol{\psi}^T},$$

which is the negative of the Hessian of $\ell(\boldsymbol{\psi})$. Then, finding a zero for the right hand side of (3.4) yields the update

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} + [I(\boldsymbol{\psi}^{(t)};\mathbf{y})]^{-1} S(\mathbf{y};\boldsymbol{\psi}^{(t)}). \tag{3.5}$$

The Newton-Raphson method has the benefit of local quadratic convergence to a solution $\boldsymbol{\psi}^*$ of (3.2), but this convergence is not guaranteed. Aside from some other computational issues (as noted in McLachlan and Krishnan (1997)), Newton-Raphson has the benefit of providing, as an estimate of the variance-covariance matrix of the solution, the inverse of the observed information matrix, $[I(\boldsymbol{\psi}^*;\mathbf{y})]^{-1}$. Thus, standard error estimates, confidence intervals, and inference procedures are readily available.

One may also implement a quasi-Newton method by replacing $I(\boldsymbol{\psi}^{(t)};\mathbf{y})$ in (3.5) by A, an approximation to the negative Hessian matrix. This yields the update

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} + A^{-1} S(\mathbf{y};\boldsymbol{\psi}^{(t)}).$$

While evaluation of the Hessian is avoided at each iteration, yielding a lower cost of computation, some drawbacks with this method are that the local quadratic convergence of the regular Newton-Raphson method is lost, convergence is not guaranteed, and erratic estimates for $\ell(\boldsymbol{\psi})$ may be obtained if a poor value of A is used.

One final Newton-type method is Fisher's method of scoring. This method replaces $I(\boldsymbol{\psi}^{(t)};\mathbf{y})$ in (3.5) by the expected (Fisher) information matrix,

$$\begin{aligned} I_f(\boldsymbol{\psi}) &= \mathrm{E}_{\boldsymbol{\psi}}\{S(\mathbf{Y};\boldsymbol{\psi})[S(\mathbf{Y};\boldsymbol{\psi})]^{\mathrm{T}}\} \\ &= -\mathrm{E}_{\boldsymbol{\psi}}\{I(\boldsymbol{\psi};\mathbf{Y})\}, \end{aligned}$$

evaluated at the current fit $\boldsymbol{\psi}^{(t)}$ for $\boldsymbol{\psi}$. This yields the update

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} + [I_f(\boldsymbol{\psi}^{(t)})]^{-1} S(\mathbf{y};\boldsymbol{\psi}^{(t)}).$$

Another version of Fisher scoring uses the empirical information matrix,

$$I_e(\boldsymbol{\psi}^{(t)}) = \sum_{i=1}^{n} S(\mathbf{y}_i;\boldsymbol{\psi}^{(t)})[S(\mathbf{y}_i;\boldsymbol{\psi}^{(t)})]^{\mathrm{T}} - \frac{1}{n}\sum_{i=1}^{n} S(\mathbf{y}_i;\boldsymbol{\psi}^{(t)})[S(\mathbf{y}_i;\boldsymbol{\psi}^{(t)})]^{\mathrm{T}},$$

yielding the update

$$\boldsymbol{\psi}^{(t+1)} = \boldsymbol{\psi}^{(t)} + [I_e(\boldsymbol{\psi}^{(t)})]^{-1} S(\mathbf{y}; \boldsymbol{\psi}^{(t)}).$$

With both methods, one is relegated to local linear convergence and convergence is again not guaranteed.

3.1.2. EM Algorithms

As seen in the previous section, Newton methods can provide relatively 'speedy' convergence, but this convergence is not ensured and calculations like inverting the Hessian may be rather difficult to perform. An alternative is the use of Expectation-Maximization (EM) algorithms, which were popularized in the mixture modeling literature after the article by Dempster et al. (1977). We will focus on developing an EM algorithm for the mixture case, but it should be noted that this algorithm is one member in a much larger class of algorithms (see McLachlan and Krishnan (1997) and McLachlan and Peel (2000) for a discussion).

We construct an EM algorithm for mixtures by first introducing the indicator random variable

$$\mathbf{Z}_{i,j} = \mathtt{I}\{\text{observation } i \text{ belongs to component } j\},$$

for $i = 1, \ldots, n$ and $j = 1, \ldots, k$ such that $\mathtt{I}\{\cdot\}$ is the indicator function. We refer to the measurements, $\mathbf{Y}$, from earlier as the *incomplete* or *observed data* and $(\mathbf{Y}, \mathbf{Z})$ as the *complete data*. Here we use $\mathbf{Y}$ and $\mathbf{Z}$ to denote all of the $\mathbf{Y}_i$'s and $\mathbf{Z}_{i,j}$'s, respectively.

The *observed data log likelihood* is simply $\ell(\boldsymbol{\psi})$ from (3.1), but it will be denoted as $\ell_o(\boldsymbol{\psi})$ when the meaning is not made clear by the context. The *complete data log likelihood* is given by

$$\begin{aligned}\ell_c(\boldsymbol{\psi}) &= \log \prod_{i=1}^{n} \prod_{j=1}^{k} [\lambda_j g(\mathbf{y}_i; \boldsymbol{\theta}_j)]^{\mathbf{Z}_{i,j}} \\ &= \sum_{i=1}^{n} \sum_{j=1}^{k} \mathbf{Z}_{i,j} \log[\lambda_j g(\mathbf{y}_i; \boldsymbol{\theta}_j)].\end{aligned}$$

$\ell_c(\boldsymbol{\psi})$ is introduced to deal with the intractability of maximizing $\ell_o(\boldsymbol{\psi})$ with respect to $\boldsymbol{\psi}$. With the formal notation defined, we now construct an EM algorithm for mixture models.

Algorithm 3.1 (**EM Algorithm**).

1. *Given a fixed* $\boldsymbol{\psi}^{(t)}$ *at the* t^{th} *iteration,* $t = 0, 1, \ldots$, *calculate*

$$\mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)}) \triangleq E_{\boldsymbol{\psi}^{(t)}}[\ell_c(\boldsymbol{\psi}) | \, \boldsymbol{Y} = \boldsymbol{y}]. \tag{3.6}$$

This step is referred to as the Expectation Step or E-Step.

2. *Find*
$$\boldsymbol{\psi}^{(t+1)} = \arg\max_{\boldsymbol{\psi}} \mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)}),$$
which implies
$$\mathcal{Q}(\boldsymbol{\psi}^{(t+1)}; \boldsymbol{\psi}^{(t)}) \geq \mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)})$$
for all $\boldsymbol{\psi} \in \Psi$. *This step is referred to as the Maximization Step or M-Step.*
3. *Iterate until a stopping criterion is attained. The final estimate obtained will be denoted by* $\hat{\boldsymbol{\psi}}$.

Notice $\mathbf{Z}_{i,j} \sim Bern(\lambda_j)$, where $Bern(\lambda_j)$ is taken to mean the Bernoulli distribution with rate of success λ_j, and $\mathbf{Z}_{i,j}$ is independent of $\mathbf{Y}_{i^*}$ for all $i^* \neq i$. Since $\mathrm{E}_{\boldsymbol{\psi}^{(t)}}$ is a linear functional, the right hand side of (**??**) and (3.6) allows us to replace $\mathbf{Z}_{i,j}$ by

$$\mathrm{E}_{\boldsymbol{\psi}}[\mathbf{Z}_{i,j}|\mathbf{Y} = \mathbf{y}] = \frac{\lambda_j g(\mathbf{y}_i; \boldsymbol{\theta}_j)}{\sum_{l=1}^{k} \lambda_l g(\mathbf{y}_i; \boldsymbol{\theta}_l)},$$

which follows from an application of Bayes' rule and the law of total probability. Thus, when provided the estimate $\boldsymbol{\psi}^{(t)}$, we get

$$\mathbf{Z}_{i,j}^{(t)} = \frac{\lambda_j^{(t)} g(\mathbf{y}_i; \boldsymbol{\theta}_j^{(t)})}{\sum_{l=1}^{k} \lambda_l^{(t)} g(\mathbf{y}_i; \boldsymbol{\theta}_l^{(t)})}.$$

Also note in the E-Step, as stressed in Flury and Zoppè (2000), is that the expectation of the complete data log likelihood is conditioned on the observed data and it does not strictly replace missing data by their conditional expectations.

As can be seen, the structure for an EM algorithm is rather simple and thus programming is relatively easy. While there have been some technical issues about the Dempster et al. (1977) article addressed over the years (such as convergence results noted by Wu (1983)), we will discuss a couple of issues concerning implementation of Algorithm 3.1.

One issue concerns selection of the *initial values* ($\boldsymbol{\psi}^{(0)}$). Due to the multimodality in the mixture likelihood, there are multiple local maxima and in some cases, a poor choice of $\boldsymbol{\psi}^{(0)}$ can lead to the sequence of EM estimates diverging. Due to such features, it is strongly advocated to start EM algorithms from many different initial values. We will use a simple binning procedure to determine hyperparameters for distributions used in random generation of the starting values. For reviews of possible options for starting values, see McLachlan and Krishnan (1997) or McLachlan and Peel (2000).

Another issue concerns the *stopping criterion*. Usually an EM algorithm is run until

$$\ell_o(\boldsymbol{\psi}^{(t+1)}) - \ell_o(\boldsymbol{\psi}^{(t)}) < \epsilon, \tag{3.7}$$

or, when given a norm $\|\cdot\|$ on $\boldsymbol{\Psi}$, until

$$\|\boldsymbol{\psi}^{(t+1)} - \boldsymbol{\psi}^{(t)}\| < \epsilon$$

for some $\epsilon > 0$ chosen arbitrarily small. Schafer (1997) discusses the stopping criterion

$$\frac{|\psi_l^{(t+1)} - \psi_l^{(t)}|}{\psi_l^{(t)}} < \epsilon,$$

for $l = 1, 2, \ldots, r$, though this method fails when $\psi_l^{(t)} \approx 0$. Regardless, EM algorithms converge linearly, which can be very slow at times. An inappropriate stopping criterion may cause one to claim convergence too soon. Certain methods, such as an Aitken-based acceleration technique, may be implemented to alleviate some of the difficulty with the slow rate of convergence (see Lindsay (1995) for a discussion). We use the method in (3.7) as our stopping criterion.

Numerous EM-type algorithms can be found in the literature (see McLachlan and Krishnan (1997) and McLachlan and Peel (2000) for references). A useful extension of the EM algorithm is the Expectation / Conditional-Maximization (ECM) algorithm of Meng and Rubin (1993). Consider a partition of $\boldsymbol{\Psi}$, say $\boldsymbol{\Psi} = (\boldsymbol{\Psi}_1^{\mathrm{T}}, \boldsymbol{\Psi}_2^{\mathrm{T}}, \ldots, \boldsymbol{\Psi}_s^{\mathrm{T}})^{\mathrm{T}}$, such that $s \leq p$.

Algorithm 3.2 (ECM Algorithm).

1. *For a given $\boldsymbol{\psi}^{(t)}$, $t = 0, 1, \ldots$, calculate*

$$\mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)}) \triangleq E_{\boldsymbol{\psi}^{(t)}}[\ell_c(\boldsymbol{\psi})|\, \boldsymbol{Y} = \boldsymbol{y}].$$

   $\boldsymbol{\psi}^{(0)}$ will be a specified initial value. This E-Step is the same step as in the EM algorithm of Algorithm 3.1.
2. *For each $i = 1, 2, \ldots, s$, calculate*

$$\boldsymbol{\psi}_i^{(t+1)} = \arg\max_{\boldsymbol{\psi}_i} \mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)}),$$

   where $\boldsymbol{\psi}_{i_1}$ is fixed at $\boldsymbol{\psi}_{i_1}^{(t)}$ for all $i_1 > i$ and $\boldsymbol{\psi}_{i_2}$ is fixed at $\boldsymbol{\psi}_{i_2}^{(t+1)}$ for all $i_2 < i$. These steps are referred to as the Conditional Maximization-Steps or CM-Steps.
3. *Iterate until a stopping criterion is attained. The final estimate obtained will be denoted by $\hat{\boldsymbol{\psi}}$.*

There is also a multicycle ECM algorithm as given in Liu and Rubin (1994). This algorithm incorporates an additional E-Step between some or all of the CM-Steps.

3.1.3. MM Algorithms and Adaptive Barrier Methods

The EM algorithms of the previous section are special cases of MM algorithms, which are prescriptions for constructing such optimization algorithms. Since we present the EM algorithm in the context of maximization, MM stands for minorize / maximize. When in the context of minimization, MM stands for maximize / minorize. For our setting, MM algorithms operate by creating a

surrogate function (which we denote by h) to drive an *objective function* uphill. In the context of our mixture model, $h(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)})$ minorizes $\ell_o(\boldsymbol{\psi})$ at $\boldsymbol{\psi}^{(t)}$ provided

$$\ell_o(\boldsymbol{\psi}^{(t)}) \quad = \quad h(\boldsymbol{\psi}^{(t)}; \boldsymbol{\psi}^{(t)})$$

and

$$\ell_o(\boldsymbol{\psi}) \quad \geq \quad h(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)}) \text{ for all } \boldsymbol{\psi}.$$

After choosing a minorizing function, we then maximize it. In the EM setting, the minorizing function at $\boldsymbol{\psi}^{(t)}$ (shifted by a constant) is $\mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)})$.

We do not go into much detail about MM algorithms here, but one may refer to Lange (2004) or Hunter and Lange (2004) for discussion. Our brief definition of MM algorithms provides a segue from EM algorithms to adaptive barrier methods, which are used in constrained optimization.

Suppose we have an ECM setting where in one of the CM-Steps

$$\begin{aligned} \boldsymbol{\psi}_l^{(t+1)} &= \arg\max_{\boldsymbol{\psi}_l} \mathcal{Q}(\boldsymbol{\psi}; \boldsymbol{\psi}^{(t)}) \\ &= \arg\max_{\boldsymbol{\psi}_l} m(\boldsymbol{\psi}_l), \end{aligned}$$

where $\boldsymbol{\psi}_l \in \mathbb{R}^{r^*}$ for some $r^* < r$, the total number of parameters. In other words, we focus only on the portion of the parameter vector over which we are actually maximizing. Suppose further that $m(\boldsymbol{\psi}_l)$ is twice continuously differentiable subject to the linear inequality constraints $c_i - u_i^{\mathrm{T}} \boldsymbol{\psi}_l \geq 0$ for $1 \leq i \leq n$. Provided $\boldsymbol{\psi}_l^{(t+1)}$ is in the interior of $\boldsymbol{\Psi}_l$, maximization of $m(\boldsymbol{\psi}_l)$ can be transferred to the surrogate function

$$h(\boldsymbol{\psi}_l; \boldsymbol{\psi}_l^{(t)}) = -m(\boldsymbol{\psi}_l) + \mu \sum_{i=1}^{n} [(c_i - u_i^{\mathrm{T}} \boldsymbol{\psi}_l^{(t)}) \log(c_i - u_i^{\mathrm{T}} \boldsymbol{\psi}_l) - u_i^{\mathrm{T}} \boldsymbol{\psi}_l],$$

where $\mu > 0$ is a barrier constant. The barrier function (the portion of the surrogate function involving μ) forces $\boldsymbol{\psi}_l^{(t+1)}$ to remain within the interior of the *feasible region* (i.e., the region satisfying the constraints).

As Lange (1999) points out, it is impossible to maximize $h(\boldsymbol{\psi}_l; \boldsymbol{\psi}_l^{(t)})$ explicitly in most problems, so maximization using a quadratic approximation is suggested. However, it is often sufficient to perform a few steps of Newton-Raphson, thus avoiding the seemingly more complex quadratic approximation method. Further details on adaptive barrier methods in convex programming may also be found in Lange (1994).

3.2. Bayesian Methods

A Bayesian approach can be taken for estimation in mixture models provided a *proper prior* is used (i.e., a prior that sums or integrates to a finite value for the

discrete and continuous case, respectively). With the advancement in computing power, developments in Markov Chain Monte Carlo (MCMC) algorithms have made Bayesian analyses an appealing method for analyzing mixture models. McLachlan and Peel (2000) provide many references to Bayesian mixture analysis. The discussion we provide below will be from the perspective where the data are continuous, but the discrete case is analogous.

Let $L_o(\boldsymbol{\psi})$ and $L_c(\boldsymbol{\psi})$ denote the observed data likelihood and complete data likelihood (the antilogarithms of $\ell_o(\boldsymbol{\psi})$ and $\ell_c(\boldsymbol{\psi})$, respectively). Let $\mathbf{z}$ denote the realization of the component indicator random variable $\mathbf{Z}$. Denote the proper prior density for $\boldsymbol{\psi}$ as $\pi(\boldsymbol{\psi})$ and the conditional density for $\mathbf{Z}$ given $\Psi = \boldsymbol{\psi}$ as $\pi(\mathbf{z}; \boldsymbol{\psi})$. The posterior density of $\boldsymbol{\psi}$ is then given by

$$\begin{aligned} p(\boldsymbol{\psi}; \mathbf{y}) &= \frac{\pi(\boldsymbol{\psi}) L_o(\boldsymbol{\psi})}{\int_{\boldsymbol{\psi} \in \Psi} \pi(\boldsymbol{\psi}) L_o(\boldsymbol{\psi}) d\boldsymbol{\psi}} \\ &= \frac{\sum_{\mathbf{z}} \pi(\mathbf{z}; \boldsymbol{\psi}) \pi(\boldsymbol{\psi}) L_c(\boldsymbol{\psi})}{\int_{\boldsymbol{\psi} \in \Psi} \sum_{\mathbf{z}} \pi(\mathbf{z}; \boldsymbol{\psi}) \pi(\boldsymbol{\psi}) L_c(\boldsymbol{\psi}) d\boldsymbol{\psi}} \\ &= K^{-1} \sum_{\mathbf{z}} \pi(\mathbf{z}; \boldsymbol{\psi}) \pi(\boldsymbol{\psi}) L_c(\boldsymbol{\psi}), \end{aligned}$$

where K denotes a normalizing constant. Now we are treating the mixture parameters as random variable quantities. We partition $\boldsymbol{\Psi}$ and denote the (independent) prior distributions on Λ_{k-1} and Θ^k by $\Pi_\Lambda(\boldsymbol{\lambda})$ and $\Pi_\Theta(\boldsymbol{\theta})$, respectively. For the prior on the mixing proportions, we will always use $\Pi_\Lambda(\boldsymbol{\lambda}) = Dir_k(\boldsymbol{\alpha})$, such that $Dir_k(\boldsymbol{\alpha})$ is taken to mean the Dirichlet distribution with parameter vector $\boldsymbol{\alpha} = (\alpha_1, \alpha_2, \ldots, \alpha_k)^{\mathrm{T}}$, where $\alpha_j > 0$ for all $j = 1, \ldots, k$. We use these priors in outlining two MCMC algorithms (in a mixture context) that are common for posterior simulation: Gibbs samplers and Metropolis-Hastings algorithms.

Using a Gibbs sampler (Geman and Geman (1984)), we may simulate from each element of $\boldsymbol{\psi}$ by conditioning on the current values of the other elements in $\boldsymbol{\psi}$. With the formal notation defined, we can now construct a Gibbs sampler for mixture models.

Algorithm 3.3 (**Gibbs Sampler**).

1. *Choose initial values* $\boldsymbol{\psi}^{(0)}$ *and* $\boldsymbol{Z}^{(0)}$.
2. *For a given* t, *such that* $t = 1, 2, \ldots$, *simulate*

$$\boldsymbol{\lambda}^{(t)} \sim Dir_k\left(\boldsymbol{\alpha} + \sum_{i=1}^{n} \boldsymbol{Z}_i^{(t-1)}\right)$$

and

$$\boldsymbol{\theta}_j^{(t)} \sim \Pi_\Theta(\boldsymbol{\theta}; \boldsymbol{Z}_j^{(t-1)})$$

for all $j = 1, \ldots, k$. *Here,* $\boldsymbol{Z}_i$ *and* $\boldsymbol{Z}_j$ *are vectors denoting the* i^{th} *row and* j^{th} *column of* $\boldsymbol{Z}$, *respectively, and* $\Pi_\Theta(\boldsymbol{\theta}; \boldsymbol{Z}_j^{(t-1)})$ *is the conditional distribution of* $\boldsymbol{\theta}$ *given the previous iteration's value of* $\boldsymbol{Z}_j$.

3. *Simulate*
$$\boldsymbol{Z}_i^{(t)} \sim Mult_k(1, \boldsymbol{\vartheta}_i^{(t)}),$$
where $Mult_k(1, \boldsymbol{\vartheta}_i^{(t)})$ *is taken to mean the multinomial distribution consisting of one draw from* k *bins with probability of success vector*
$$\boldsymbol{\vartheta}_i^{(t)} = \left(\frac{\lambda_1^{(t)} g(y_i; \boldsymbol{\theta}_1^{(t)})}{\sum_{l=1}^k \lambda_l^{(t)} g(y_i; \boldsymbol{\theta}_l^{(t)})}, \ldots, \frac{\lambda_k^{(t)} g(y_i; \boldsymbol{\theta}_k^{(t)})}{\sum_{l=1}^k \lambda_l^{(t)} g(y_i; \boldsymbol{\theta}_l^{(t)})} \right)^T.$$
4. *Increment* t *and repeat steps 2 and 3.*

As for a Metropolis-Hastings algorithm (Metropolis et al. (1953) and Hastings (1970)), a little more programming is usually necessary since there is the requirement of a *proposal density* ($q(\boldsymbol{\psi}^*; \boldsymbol{\psi})$) to effectively search the entire parameter space. Many times, this density is chosen to be symmetric (i.e., $q(\boldsymbol{\psi}^*; \boldsymbol{\psi}) = q(\boldsymbol{\psi}; \boldsymbol{\psi}^*)$), but as Gill (2002) points out, this is not necessary. The decision about whether we accept a value, $\boldsymbol{\psi}^*$, from this proposal density will be based on the *acceptance ratio*,
$$a(\boldsymbol{\psi}^*; \boldsymbol{\psi}) = \frac{q(\boldsymbol{\psi}; \boldsymbol{\psi}^*) p(\boldsymbol{\psi}^*)}{q(\boldsymbol{\psi}^*; \boldsymbol{\psi}) p(\boldsymbol{\psi})}.$$

With the formal notation defined, we can now construct a Metropolis-Hastings algorithm for mixture models.

Algorithm 3.4 (**Metropolis-Hastings Algorithm**).

1. *Choose initial values* $\boldsymbol{\psi}^{(0)}$. *Let* $t = 0$.
2. *Sample* $\boldsymbol{\psi}^*$ *from* $q(\boldsymbol{\psi}^*; \boldsymbol{\psi}^{(t)})$.
3. *Generate* $u \sim Unif(0, 1)$, *such that* $Unif(0, 1)$ *is taken to mean the uniform distribution over the interval (0,1).*
4. *Set*
$$\boldsymbol{\psi}^{(t+1)} = \begin{cases} \boldsymbol{\psi}^*, & a(\boldsymbol{\psi}^*; \boldsymbol{\psi}^{(t)}) > u; \\ \boldsymbol{\psi}^{(t)}, & a(\boldsymbol{\psi}^*; \boldsymbol{\psi}^{(t)}) \leq u. \end{cases}$$
5. *Increment* t *and repeat steps 2 through 4.*

Once we have an MCMC sample from the posterior, we may perform inference for the parameters. This is in contrast to likelihood methods, that give only maximum likelihood estimates and an estimate of its sampling distribution variance-covariance matrix.

We should note some issues when implementing these and other MCMC methods as found in Robert and Casella (2004). For instance, choosing a proposal density $q(\boldsymbol{\psi}^*; \boldsymbol{\psi})$ may require one to incorporate some sort of tuning parameter (see Chib and Greenberg (1995) and Cappè and Robert (2000)). There are also practical issues such as thinning the chain, burn-in, and selecting initial values. A major problem in using MCMC methods to estimate parameters in the mixture setting is label switching, which will be addressed later.

3.3. Number of Components

Determining the number of components for (2.1) is still a major contemporary issue in mixture modeling. We will address here some of the techniques used in assessing the number of components when this is not known *a priori*.

3.3.1. Likelihood Approaches

From a likelihood perspective, we consider testing

$$\begin{aligned} H_0 &: k = k_0 \\ H_1 &: k = k_0 + 1 \end{aligned} \tag{3.8}$$

for some positive integer k_0. Letting $\hat{\boldsymbol{\psi}}_1$ and $\hat{\boldsymbol{\psi}}_2$ denote the MLEs of $\boldsymbol{\psi}$ calculated under H_0 and H_1, respectively, we could consider the likelihood ratio test (LRT) statistic

$$-2\log\Delta = 2\{\ell(\hat{\boldsymbol{\psi}}_1) - \ell(\hat{\boldsymbol{\psi}}_0)\}. \tag{3.9}$$

It is well known that standard regularity conditions do not hold in the setting of (3.8) and thus the asymptotic distribution of (3.9) is not the usual chi-squared distribution (see Aitkin and Rubin (1985) and Lindsay (1995) for a discussion). However, model selection techniques are still used in assessing the overall number components as simulations have indicated relatively good empirical results (see McLachlan and Peel (2000) for references). We recommend not using these techniques solely in determining the number of components of a mixture model, but rather to give further supporting evidence to the number selected based on another method, such as a bootstrapping technique (to be discussed later).

Four common model selection criteria are Akaike's information criterion (AIC) of Akaike (1973), the Bayesian information criterion (BIC) of Schwarz (1978), the Integrated Completed Likelihood (ICL) of Biernacki et al. (2000), and the consistent AIC (CAIC) of Bozdogan (1987). Given an estimate $\hat{\boldsymbol{\psi}}$, the form of these criteria are, respectively,

$$\begin{aligned} \text{AIC} &= \ell(\hat{\boldsymbol{\psi}}) - r \\ \text{BIC} &= \ell(\hat{\boldsymbol{\psi}}) - \frac{r}{2}\log(n) \\ \text{ICL} &= \text{BIC} - \sum_{j=1}^{k}\hat{\lambda}_j\log(\hat{\lambda}_j) \\ \text{CAIC} &= \ell(\hat{\boldsymbol{\psi}}) - \frac{r}{2}(\log(n)+1), \end{aligned}$$

where $r = kq + k - 1$ is the number of parameters in the mixture setting. These values are calculated for a reasonable range of components and then the maximum of these values (for each criterion) corresponds to the number of components selected by that criterion.

As an alternative to penalized likelihood methods, Chen and Kalbfleisch (1996) present a penalized minimum-distance estimate. They argue that the penalized likelihood approach tends to produce a fit with fewer components. However, it is unknown whether or not this approach produces a consistent estimate of the number of mixture components. Hence, they use the penalized minimum-distance estimate, which they show to be consistent for the number of mixture components as well as the mixing distribution. Their method is used for any of the common distances, such as the Kolmogorov-Smirnov distance, the Cramer-von Mises distance, and the Kullback-Liebler information. While the last is not symmetric, it can be viewed as a distance and used when two distributions have common support. In fact, one may use a symmetrized version of the Kullback-Liebler distance to avoid the symmetry issue. The interested reader may refer to Chen and Kalbfleisch (1996) for applications of the penalized minimum-distance method.

A commonly employed method in determining the number of components is a bootstrapping scheme proposed by McLachlan (1987). The algorithm is an attempt to approximate the null distribution of the LRT statistic values given in (3.9). We outline the algorithm for this parametric bootstrapping scheme using an EM algorithm as follows:

Algorithm 3.5 (**Parametric Bootstrapping the LRTs for Number of Components**).

1. *Fit a mixture model with k_0 and k_0+1 components to the data, $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$, which leads to the EM estimates $\hat{\boldsymbol{\psi}}_1$ and $\hat{\boldsymbol{\psi}}_2$, respectively.*
2. *Calculate the (observed) log likelihood ratio statistic in (3.9). Denote this value by Ξ_{obs}.*
3. *Simulate a data set of size n from the null distribution (the model with k_0 components). Call this sample $\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_n^*$.*
4. *Fit a mixture model with k_0 and $k_0 + 1$ components to the simulated data and calculate the corresponding 'bootstrap' log likelihood ratio statistic. Denote this value by Ξ^*.*
5. *Repeat steps 3 and 4 B times to generate the bootstrap sampling distribution of the likelihood ratio statistic, $\Xi_1^*, \Xi_2^*, \ldots, \Xi_B^*$.*
6. *Compute the bootstrap p-value as*

$$p_B = \frac{1}{B} \sum_{i=1}^{B} I\{\Xi_{obs} \geq \Xi_i^*\}.$$

Algorithm 3.5 is implemented by first testing 1 versus 2 components. A value of p_B is obtained for this test and if it is lower than some significance level α, then claim statistical significance and proceed to test 2 versus 3 components. If not, stop and claim that there is not statistically significant evidence for a 2-component fit. Proceed in this manner until you fail to reject the null hypothesis.

Exact theoretical results for testing (3.8) have been obtained in numerous special cases. As Lindsay (1995) points out, some of these testing scenarios yield

limiting distributions that either resemble mixtures of chi-squared distributions of different degrees of freedom (called a *chi-bar-squared* distribution) or can, in fact, be shown to be a chi-bar-squared distribution. One special case is when $k_0 = 1$ in (3.8). Lindsay (1995) shows the limiting distribution for $-2\log\Delta$ in this case is

$$\frac{1}{2}\chi_0^2 + \frac{1}{2}\chi_1^2, \tag{3.10}$$

where χ_p^2 denotes the chi-squared distribution with p degrees of freedom.

Notice that (3.10) is just a linear combination of chi-squares. Because of this fact, there is no guarantee that the parametric bootstrap outlined in Algorithm 3.5 will give a good approximation. An example where a statistic is asymptotically distributed as a linear combination of chi-squares and the parametric bootstrap approximation fails can be found in Babu (1984). While this does present theoretical difficulties, it does not appear to be an issue often encountered in practice. McLachlan and Peel (2000) present simulation results and cite many references that endorse this method by assessing the accuracy of p_B as well as the overall power of the test.

3.3.2. Bayesian Approaches

In addition to the likelihood methods presented, there are a few Bayesian procedures concerning estimating the number of components. One method is the Dirichlet process (Ferguson (1973)). For a parametric mixture model, rewrite (2.1) as

$$f(\mathbf{y}_i;\boldsymbol{\psi}) = \sum_{j=1}^{k}\lambda_j g(\mathbf{y}_i;\boldsymbol{\theta}_j) = \int g(\mathbf{y}_i;\boldsymbol{\theta})dP(\boldsymbol{\theta}),$$

where the mixing distribution P is defined as $\sum_{j=1}^{k}\delta(\boldsymbol{\theta}_j)$. Here, $\delta(\cdot)$ represents the Dirac measure on the parameter space Θ, meaning that P is a discrete distribution that puts mass λ_j on $\boldsymbol{\theta}_j$. Since the number of components is now random, the problem can be thought of as selecting a model out of the set of all possible mixture distributions on Θ. Thus, it is necessary to specify some sort of prior on this set. One way of accomplishing this is by implementing the Dirichlet process to obtain the prior on the set of all distributions on Θ.

The focus is on P, which is a distribution on Θ drawn from the Dirichlet process with parameter α, such that α is a finite measure on Θ. While the details of the Dirichlet process (Ferguson (1996)) are beyond the scope of this work, an easier way to think about the Dirichlet process is what is referred to as Sethuraman's representation:

Algorithm 3.6 (**Sethuraman's Representation of the Dirichlet Process**)**.**

1. *Let $\bar{\alpha}$ be the probability measure $\alpha/\alpha(\Theta)$ and take $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots$ as independent and identically distributed from $\bar{\alpha}$.*

2. *Take* $\gamma_1, \gamma_2, \ldots$ *as independent and identically distributed from* $Beta(1, \alpha(\Theta))$*, chosen independently of* $\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots$*, such that* $Beta(a, b)$ *is taken to mean the beta distribution with shape parameters* a *and* b*.*
3. *Define*

$$\begin{aligned}
\gamma_1^* &= \gamma_1 \\
\gamma_2^* &= \gamma_2(1 - \gamma_1^*) \\
&\vdots \\
\gamma_k^* &= \gamma_k\left(1 - \sum_{j=1}^{k-1} \gamma_j^*\right) \\
&\vdots
\end{aligned}$$

4. *Let* $P = \sum_{j=1}^{\infty} \gamma_j^* \delta(\boldsymbol{\theta}_j)$*, so that* P *puts mass* γ_j^* *at* $\boldsymbol{\theta}_j$*.*

Sethuraman (1994) showed that P is a realization from the Dirichlet process with parameter α. In addition to Algorithm 3.6, Escobar (1994) presented a way to sample from the posterior when a Dirichlet prior is used with a location mixture of normals.

Green (1995) proposed a framework for constructing a reversible jump MCMC in order to "jump" between parameter subspaces of varying dimensionality. This is appealing for Bayesian model determination because now prior information can be placed on the number of components in the model (as well as the component parameters) and provide effective exploration of the varying dimensions of parameter subspaces. Since k is now a parameter, the parameter vector of interest becomes

$$\boldsymbol{\omega}_k = (\boldsymbol{\theta}_1^{\mathrm{T}}, \ldots, \boldsymbol{\theta}_k^{\mathrm{T}}, \lambda_1, \ldots, \lambda_{k-1}, k)^{\mathrm{T}} \in \boldsymbol{\Omega}_k,$$

such that $k \in \mathbb{N}$. The elements of $\boldsymbol{\omega}_k$ are each regarded as being drawn from an appropriately defined prior distribution. A basic sketch of the reversible jump MCMC method is as follows:

Algorithm 3.7 (**Reversible Jump MCMC**)**.**

1. *Draw the value* $\boldsymbol{\omega}_k^*$ *from the proposal density* $g(\cdot|\boldsymbol{\omega}_k)$ *and target (posterior) distribution* $\pi(\cdot)$*. (Note that* $\boldsymbol{\omega}_k^*$ *may be from a different subspace than* $\boldsymbol{\omega}_k$*.)*
2. *Let* $M = M_1 \cup M_2$ *be a countable family of move types.*
   (a) *If move* $m \in M_1$ *is attempted with destination* $\boldsymbol{\omega}_k^* \in \boldsymbol{\Omega}_k$*, then the acceptance of this sample is given by the appropriately defined acceptance probability* $\alpha_m^{(1)}(\boldsymbol{\omega}_k, \boldsymbol{\omega}_k^*)$*.*
   (b) *If move* $m \in M_2$ *is attempted with destination* $\boldsymbol{\omega}_k^* \in \boldsymbol{\Omega}_{k'}$*, such that* $k \neq k'$*, then the acceptance of this sample is given by the appropriately defined acceptance probability* $\alpha_m^{(2)}(\boldsymbol{\omega}_k, \boldsymbol{\omega}_k^*)$*.*
3. *Iterate.*

Richardson and Green (1997) detail how to construct the acceptance probabilities in the above algorithm. Also, some of the possible moves m the authors suggest include:

1. Updating any (or all) of $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k$ (an M_1-type move).
2. Updating $\lambda_1, \ldots, \lambda_{k-1}$ (an M_1-type move).
3. Splitting one of the mixture components into two components (an M_2-type move).
4. Merging two of the mixture components into one component (an M_2-type move).

Stephens (2000a) provided a birth-and-death MCMC as an alternative to the reversible jump MCMC. The birth-and-death MCMC is a marked point process which can be viewed as a continuous-time version of the reversible jump MCMC with a limited number of moves. By limiting the number of moves, the implementation of the algorithm is simplified.

A *birth* is said to occur if at time $t \in \mathbb{R}_+$ the process is at $\boldsymbol{\psi} \in \boldsymbol{\Psi}_k$ and then jumps to

$$\boldsymbol{\psi} \cup (\lambda, \boldsymbol{\theta}) \equiv \{\lambda_1(1-\lambda), \ldots, \lambda_{k-1}(1-\lambda), \lambda, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_k, \boldsymbol{\theta}\} \in \boldsymbol{\Psi}_{k+1}, \quad (3.11)$$

while a *death* (of the i^{th} component) occurs if the process jumps to

$$\boldsymbol{\psi} \backslash (\lambda_i, \boldsymbol{\theta}_i) \equiv \left\{ \frac{\lambda_1}{1-\lambda_i}, \ldots, \frac{\lambda_{i-1}}{1-\lambda_i}, \frac{\lambda_{i+1}}{1-\lambda_i}, \ldots, \frac{\lambda_{k-1}}{1-\lambda_i}, \right.$$
$$\left. \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_{i-1}, \boldsymbol{\theta}_{i+1}, \ldots, \boldsymbol{\theta}_k \right\} \in \boldsymbol{\Psi}_{k-1}. \quad (3.12)$$

The overall rate that births occur is $\beta(\boldsymbol{\psi})$ and the point $(\lambda, \boldsymbol{\theta})$ is chosen according to some density $h(\boldsymbol{\psi}; \lambda, \boldsymbol{\theta})$. Also, each point $(\lambda_i, \boldsymbol{\theta}_i)$ dies independently of the others according to a Poisson process with rate $\delta_j(\boldsymbol{\psi})$, where the overall death rate is $\delta(\boldsymbol{\psi}) = \sum_{j=1}^{k} \delta_j(\boldsymbol{\psi})$. A basic sketch of the birth-and-death MCMC is as follows:

Algorithm 3.8 (Birth-and-Death MCMC). *Set $t = 0$ where t is the time of a birth or death. For $\nu = 0, 1, \ldots, V$, run the following until $t > \nu + 1$:*

1. *Define the birth rate $\beta(\boldsymbol{\psi})$ as a constant and calculate the death rate $\delta_j(\boldsymbol{\psi})$ of each component.*
2. *Calculate the overall death rate $\delta(\boldsymbol{\psi})$ and simulate $u \sim Unif(0, 1)$.*
3. *Update t with*
$$t - \frac{\log(u)}{\beta(\boldsymbol{\psi}) + \delta(\boldsymbol{\psi})}.$$
4. *Simulate the type of jump.*
   (a) *If the jump is a "birth", then simulate the new component $(\lambda, \boldsymbol{\theta})$ from the density $h(\boldsymbol{\psi}; \lambda, \boldsymbol{\theta})$ and update $\boldsymbol{\psi}$ as in (3.11).*
   (b) *If the jump is a "death", then select the new component $(\lambda_j, \boldsymbol{\theta}_j)$ to die with probability $\delta_j(\boldsymbol{\psi})/\delta(\boldsymbol{\psi})$ and update $\boldsymbol{\psi}$ as in (3.12).*

5. *Run I iterations of an MCMC sampler according to the current parameter space.*

3.4. Standard Errors

After generating an MCMC sample using the procedures discussed in Section 3.2, posterior standard deviations are easily obtained. With likelihood methods, it is possible to obtain standard error estimates by using the inverse of the observed information matrix when implementing a Newton-type method. However, this may be computationally burdensome. An alternative way to report standard errors in the likelihood setting is by implementing a parametric bootstrap (Efron and Tibshirani (1993)). Efron and Tibshirani (1993) claim the parametric bootstrap should provide similar estimates to the standard errors compared to the method involving the information matrix. The development of this procedure has become useful for the mixture case as well. We outline the algorithm for a parametric bootstrapping scheme in the mixture setting using an EM algorithm as follows:

Algorithm 3.9 (**Parametric Bootstrap for Standard Errors**).

1. *Find the maximum likelihood estimate* $\hat{\boldsymbol{\psi}}$ *by implementing an EM algorithm (Algorithm 3.1) based on the values* $\boldsymbol{y}_1, \boldsymbol{y}_2, \ldots, \boldsymbol{y}_n$.
2. *Generate a bootstrap sample of size* n *from* $f(\boldsymbol{y}; \hat{\boldsymbol{\psi}})$ *and call this sample* $\boldsymbol{y}_1^*, \boldsymbol{y}_2^*, \ldots, \boldsymbol{y}_n^*$.
3. *Find the estimate* $\tilde{\boldsymbol{\psi}}$ *for the bootstrap sample by implementing an EM algorithm.*
4. *Repeat steps 2 and 3* B *times to generate the bootstrap sampling distribution* $\tilde{\boldsymbol{\psi}}^{(1)}, \tilde{\boldsymbol{\psi}}^{(2)}, \ldots, \tilde{\boldsymbol{\psi}}^{(B)}$.

After implementing Algorithm 3.9, the bootstrap variance-covariance matrix is easily computed as the sample variance-covariance matrix of the generated values $\tilde{\boldsymbol{\psi}}^{(1)}, \tilde{\boldsymbol{\psi}}^{(2)}, \ldots, \tilde{\boldsymbol{\psi}}^{(B)}$. Thus, bootstrap standard errors are readily available. However, when performing a bootstrapping procedure in the mixture setting, one must be cognizant of the label switching problem described below.

4. Identifiability

In this section, we formally define identifiability for mixture distributions. This discussion and the definition of identifiability are adopted from McLachlan and Peel (2000).

Let $\mathcal{F}_k$ denote a parametric family of k-component mixture densities as described in (2.1) and $\mathcal{F}$ the class of all such $\mathcal{F}_k$. So

$$\mathcal{F}_k = \{f_k(\mathbf{y}_i; \boldsymbol{\psi}) : \boldsymbol{\psi} \in \Psi\} \quad \text{and} \quad \mathcal{F} = \bigcup_{k \in \mathbb{N}} \mathcal{F}_k.$$

Permuting the component labels of the mixture density results in $\mathcal{F}$ being nonidentifiable in Ψ. We formalize this concept as a definition.

Definition 4.1 (**Identifiability**). *Consider*

$$f_k(\boldsymbol{y}_i;\boldsymbol{\psi}) = \sum_{j=1}^{k} \lambda_j g(\boldsymbol{y}_i;\boldsymbol{\theta}_j)$$

and

$$f_{k^*}(\boldsymbol{y}_i;\boldsymbol{\psi}^*) = \sum_{j=1}^{k^*} \lambda_j^* g(\boldsymbol{y}_i;\boldsymbol{\theta}_j^*),$$

which are both members of the class $\mathcal{F}$. $\mathcal{F}$ is said to be identifiable for $\boldsymbol{\psi} \in \Psi$ if $f_k(\boldsymbol{y}_i;\boldsymbol{\psi}) = f_{k^*}(\boldsymbol{y}_i;\boldsymbol{\psi}^*)$ a.e. $\nu[\boldsymbol{y}_i]$ if and only if (i) $k = k^*$; (ii) under permutation of the component labels, $\lambda_j = \lambda_j^*$ and $g(\boldsymbol{y}_i;\boldsymbol{\theta}_j) = g(\boldsymbol{y}_i;\boldsymbol{\theta}_j^*)$ a.e. $\nu[\boldsymbol{y}_i]$ for all $j = 1,\ldots,k$ and (iii) $\lambda_j > 0$ and the $\boldsymbol{\theta}_j$ are distinct for all j. Here, $\nu[\cdot]$ is the underlying measure on $\mathbb{R}^q$ for $g(\boldsymbol{y}_i;\boldsymbol{\theta})$.

Definition 4.1 states that no element of $\mathcal{F}$ can arise in two different ways except by trivial means, such as letting some $\lambda_j = 0$ or splitting a component by letting $\boldsymbol{\theta}_{j_1} = \boldsymbol{\theta}_{j_2}$.

4.1. Label Switching

In Section 3, we saw possible estimation methods used in mixture modeling. These methods included a parametric bootstrap using EM algorithms to obtain standard error estimates and Bayesian inference via MCMC samplers. During the implementation of such iterative methods, one must be cognizant of the solutions being calculated from one iteration to the next since a given mixture component cannot be extracted from the likelihood. This situation occurs because the component labels cannot be distinguished from one another due to the nonidentifiability in $\boldsymbol{\psi}$ as established in Definition 4.1. Such a permutation of the component labels as in this definition is called *label switching*.

There are numerous methods in the literature for dealing with label switching (see Jasra et al. (2005) for a review of some of these techniques). One of the easiest methods for dealing with this issue, especially when the parameters are well-separated within the parameter space, is by imposing identifiability constraints on the parameters (such as $\lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_k$). However, this method comes with caveats heavily emphasized in the literature (for instance, see McLachlan and Peel (2000) and Stephens (2000b)). For example, consider fitting a mixture with $k = 2$ components with the mixing proportions close to 0.50. Imposing the identifiability constraint on the mixing proportions clearly influences the estimates of $\boldsymbol{\theta}_1$ and $\boldsymbol{\theta}_2$, thus creating a bias. Such a situation is highlighted in Celeux et al. (2000) where they present "disturbing" results when considering the various ordering constraints on a $k = 3$ component mixture of normals using an MCMC sampler. This identifiability can be imposed after the simulations have been completed, as Stephens

(1997) demonstrates for an MCMC sample of size N by relabeling the sample $(\boldsymbol{\Psi}^{(1)}, \boldsymbol{\Psi}^{(2)}, \ldots, \boldsymbol{\Psi}^{(N)})$ and applying permutations $\pi_1, \pi_2, \ldots, \pi_N$ such that the permuted sample $(\pi_1(\boldsymbol{\Psi}^{(1)}), \pi_2(\boldsymbol{\Psi}^{(2)}), \ldots, \pi_N(\boldsymbol{\Psi}^{(N)}))$ satisfies the identifiability constraints. Since there is not always a clear choice of labeling, Richardson and Green (1997) stress post-processing the simulations under different permutations of the labels to determine an appropriate choice.

One alternative method is to consider bootstrapping in mixtures. McLachlan and Peel (2000) point out that label switching can usually be avoided by setting the EM algorithm's starting values to the maximum likelihood estimates, since EM algorithms are (generally) very dependent on the starting values.

Next, note that since the likelihood of a k-component mixture model is invariant under permutation of the component labels, it effectively has $k!$ modes. Label switching is often presented in the context of Bayesian mixture modeling since the posterior distribution will also have this property under a symmetric prior. The first Bayesian method we will consider is a decision theoretic approach as implemented in Celeux et al. (2000), Stephens (2000b), Hurn et al. (2003), and Jasra et al. (2005).

Consider estimating the parameters from the mixture model in (2.1). In a Bayesian framework, summarizing their posterior distributions will be viewed as choosing an *action*, a, from the *action space*, $\mathcal{A}$. Then, define a *loss function* $\mathcal{L} : \mathcal{A} \times \Psi \mapsto \mathbb{R}^+$. Since a will be a chosen vector of parameters, let $a = \tilde{\boldsymbol{\psi}}$. The objective is to find a value of $\tilde{\boldsymbol{\psi}}$ that minimizes the posterior expected loss, or *risk*. This results in the *Bayes estimator*, which is defined as

$$\tilde{\boldsymbol{\psi}}^* = \arg\min_{\tilde{\boldsymbol{\psi}}} \mathrm{E}_{\boldsymbol{\psi}|\mathbf{Y}}[\mathcal{L}(\tilde{\boldsymbol{\psi}}; \boldsymbol{\psi})]. \tag{4.1}$$

The expectation on the right hand side is the *risk function*, which is taken over the posterior distribution of $\boldsymbol{\psi}|\mathbf{Y}$. When considering only the class of loss functions that are invariant under permutations of $\boldsymbol{\psi}$, the Bayes estimator in (4.1) becomes unaffected by label indexes. Once an appropriate loss function has been chosen, these procedures can be summarized into the following algorithm:

Algorithm 4.1 (**Loss-Based Algorithm for Label Switching**).

1. *Fix a value of* $\tilde{\boldsymbol{\psi}}$.
2. *Generate a large number of realizations from the posterior distribution of* $\boldsymbol{\psi}|\,\boldsymbol{Y}$ *using an MCMC sampler. Call these realizations* $\boldsymbol{\psi}^{(1)}, \boldsymbol{\psi}^{(2)}, \ldots, \boldsymbol{\psi}^{(N)}$.
3. *Discard the first* M *iterations for burn-in.*
4. *Calculate the MCMC ergodic averages (called Monte Carlo risk in Stephens (2000b)) as*

$$\begin{aligned}\mathcal{R}(\tilde{\boldsymbol{\psi}}) &= E_{\boldsymbol{\psi}|\,\boldsymbol{Y}}[\mathcal{L}(\tilde{\boldsymbol{\psi}}; \boldsymbol{\psi})] \\ &= \frac{1}{N-M} \sum_{i=M+1}^{N} \mathcal{L}(\tilde{\boldsymbol{\psi}}; \boldsymbol{\psi}^{(i)}).\end{aligned}$$

5. *Find the Bayes estimator* $\tilde{\boldsymbol{\psi}}^* = \arg\min_{\tilde{\boldsymbol{\psi}}} \mathcal{R}(\tilde{\boldsymbol{\psi}})$.

This entire process hinges on the selection of an appropriate loss function, which may be quite challenging. Yet if the loss function $\mathcal{L}$ is chosen to be invariant to permutations of the component labels, then label switching will not hamper the resulting Bayesian estimates. Stephens (2000b) recommends running Algorithm 4.1 from several starting points and choosing the Bayes estimate that provides the best local optimum found.

Another procedure used within the Bayesian framework is by Chung et al. (2004), who suggest assigning as few as one observation to a component *a priori*. This amounts to using data-dependent priors where one or more observations are assigned to each component with certainty. The point is to apply enough information to break the symmetry of the likelihood and flatten the posterior density over $k! - 1$ nuisance regions, which are the duplicate modes resulting from the permutations of the components. The posterior density in the sampler will now reflect a modified likelihood function which accommodates a density where one (or more) observations were assigned to each component. The major limitation of this approach is to what extent one is willing to accept preclassifying certain observations.

5. Software

In this section, we briefly outline a few software packages capable of performing analysis of mixture models. There are many packages which specialize in fitting certain mixture models, but the packages we mention here provide a little more versatility with respect to the selection of functions they offer.

The SAS TRAJ procedure (Jones et al. (2001)) analyzes longitudinal data by fitting a mixture model. Specifically, PROC TRAJ fits semiparametric discrete mixture models to longitudinal data. Distributions available in this procedure include Bernoulli, censored normal, Poisson, and zero-inflated Poisson.

The `R` programming language has a few packages available for analyzing mixture models. The `mclust` package (Fraley and Raftery (2006)) provides model-based clustering, density estimation, discriminant analysis, and the analysis of mixtures of (multivariate) normals under various parameterizations of the component-specific variance-covariance matrices. EM algorithms are used for estimation and BIC is used for determining the number of components.

Another package available in `R` is the `mixtools` package (Young et al. (2008)). This package fits a wide array of mixture models including mixtures of (multivariate) normals, mixtures of regressions, mixtures of Poisson regressions, mixtures of logistic regressions, and mixtures of multinomials. EM algorithms are used for estimation in all of these cases and there is also a Metropolis-Hastings algorithm for the mixture of regressions setting. There are bootstrapping functions for testing the number of components as well as estimating the standard errors. There is also a stochastic semiparametric EM algorithm for estimating a nonparametric multivariate mixture model.

One commercially available software program is M*plus* (Muthèn and Muthèn (2008)). This program provides tools for mixture models, latent class analysis,

and survival mixtures, just to name a few. M*plus* has the capability of carrying out these analyses for observed variables that are continuous, censored, binary, ordinal, nominal, or any combinations of these types. M*plus* is a very flexible tool for researchers and provides many routines in addition to those for mixture analysis.

6. Conclusion

The area of finite mixture models has a rich literature demonstrating their applicability in a wide variety of fields. This article attempted to codify an overview of mixture modeling by providing an introduction to the topic, discussion of relevant issues in estimation, and outlining various algorithms. Researchers unfamiliar with mixture modeling will hopefully gain an appreciation of their utility as well as an understanding of their limitations. We hope the reader gained a greater breadth of knowledge to aid them as they proceed with more specific literature on the various tools and issues regarding mixture modeling.

References

M. Aitkin and D. B. Rubin. Estimation and hypothesis testing in finite mixture models. *Journal of the Royal Statistical Society, Series B*, 47(1):67–75, 1985.

H. Akaike. Information theory and an extension of the maximum likelihood principle. In B. N. Petrov and F. Csaki, editors, *Second International Symposium on Information Theory*, pages 267–281. Akademiai Kiado, Budapest, 1973.

T. W. Anderson. *An Introduction to Multivariate Statistical Analysis*. Wiley, New Jersey, 3rd edition, 2003.

G. J. Babu. Bootstrapping statistics with linear combinations of chi-squares as weak limit. *Sankhyā - The Indian Journal of Statistics*, 46(1):85–93, 1984.

C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725, 2000.

H. Bozdogan. Model selection and Akaike's information criterion (AIC): The general theory and its analytical extensions. *Psychometrika*, 52(3):345–370, 1987.

O. Cappè and C. P. Robert. Markov Chain Monte Carlo: 10 years and still running! *Journal of the American Statistical Association*, 95(452):1282–1286, 2000.

G. Celeux, M. Hurn, and C. P. Robert. Computational and inferential difficulties with mixture posterior distributions. *Journal of the American Statistical Association*, 95(451):957–970, 2000.

J. Chen and J. D. Kalbfleisch. Penalized minimum-distance estimates in finite mixture models. *The Candadian Journal of Statistics*, 24(2):167–175, 1996.

S. Chib and E. Greenberg. Understanding the Metropolis-Hastings algoirthm. *The American Statistician*, 49(4):327–335, 1995.

H. Chung, E. Loken, and J. L. Schafer. Difficulties in drawing inferences with finite-mixture models: A simple example with a simple solution. *The American Statistician*, 58(2):152–158, 2004.

A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

B. Efron and R. Tibshirani. *An Introduction to the Bootstrap*. Chapman & Hall, London, 1993.

M. D. Escobar. Estimating normal means with a Dirichlet process prior. *Journal of the American Statistical Association*, 89(425):268–277, 1994.

B. S. Everitt and D. J. Hand. *Finite Mixture Distributions*. Chapman & Hall, London, 1981.

T. S. Ferguson. A Bayesian analysis of some nonparametric problems. *The Annals of Statistics*, 1(2):209–230, 1973.

T. S. Ferguson. *A Course in Large Sample Theory*. Chapman & Hall/CRC, Florida, 1996.

B. Flury and A. Zoppè. Exercises in EM. *The American Statistician*, 54(3): 207–209, 2000.

C. Fraley and A. Raftery. *mclust: Model-Based Clustering / Normal Mixture Modeling*, 2006. URL http://www.stat.washington.edu/mclust. R package version 3.0-0.

S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, 1984.

J. Gill. *Bayesian Methods: A Social and Behavioral Sciences Approach*. Chapman & Hall/CRC, Florida, 2002.

P. J. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82(4):711–732, 1995.

W. K. Hastings. Monte Carlo sampling methods using Markov Chains and their applications. *Biometrika*, 57(1):97–109, 1970.

D. R. Hunter and K. Lange. MM algorithms for generalized Bradley-Terry models. *The American Statistician*, 58(1):30–37, 2004.

M. Hurn, A. Justel, and C. P. Robert. Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1):55–79, 2003.

A. Jasra, C. C. Holmes, and D. A. Stephens. Markov Chain Monte Carlo methods and the label switching problem in Bayesian mixture modeling. *Statistical Science*, 20(1):50–67, 2005.

R. A. Johnson and D. W. Wichern. *Applied Multivariate Statistical Analysis*. Prentice Hall, New Jersey, 5^{th} edition, 2002.

B. L. Jones, D. S. Nagin, and K. Roeder. A sas procedure based on mixture models for estimating developmental trajectories. *Sociological Methods & Research*, 29(3):374–393, 2001.

K. Lange. An adaptive barrier method for convex programming. *Methods and Applications of Analysis*, 1(4):392–402, 1994.

K. Lange. *Numerical Analysis for Statisticians*. Springer-Verlag, New York, 1999.

K. Lange. *Optimization*. Springer-Verlag, New York, 2004.

B. G. Lindsay. *Mixture Models: Theory, Geometry and Applications*, volume 5 of *NSF-CBMS Regional Conference Series in Probability and Statistics*. Institute of Mathematical Statistics and the American Statistical Association, 1995.

C. Liu and D. B. Rubin. The ECME algorithm: A simple extension of EM and ECM with faster monotone convergence. *Biometrika*, 81(4):633–648, 1994.

G. J. McLachlan. On bootstrapping the likelihood ratio test statistic for the number of components in a normal mixture. *Applied Statistics*, 36(3):318–324, 1987.

G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, New York, 1997.

G. J. McLachlan and D. Peel. *Finite Mixture Models*. Wiley, New York, 2000.

X. L. Meng and D. B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267–278, 1993.

N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller. Equation of state calculations by fast computing machine. *Journal of Chemical Phsyics*, 21(6):1087–1091, 1953.

B. Muthèn and L. Muthèn. *Mplus*, 2008. URL http://www.statmodel.com/. Mplus version 5.1.

K. Pearson. Contributions to the mathematical theory of evolution. *Philosophical Transactions of the Royal Society of London A*, 185:71–110, 1894.

F. Peng, R. A. Jacobs, and M. A. Tanner. Bayesian inference in mixtures-of-experts and hierarchical mixtures-of-experts models with an application to speech recognition. *Journal of the American Statistical Association*, 91(435): 953–960, 1996.

R. E. Quandt and J. B. Ramsey. Estimating mixtures of normal distributions and switching regressions. *Journal of the American Statistical Association*, 73(364):730–738, 1978.

S. Richardson and P. J. Green. On Bayesian analysis of mixtures with an unknown number of components. *Journal of the Royal Statistical Society, Series B*, 59(4):731–792, 1997.

C. P. Robert and G. Casella. *Monte Carlo Statistical Methods*. Springer-Verlag, New York, 2nd edition, 2004.

J. L. Schafer. *Analysis of Incomplete Multivariate Data*. Chapman & Hall, Florida, 1997.

G. Schwarz. Estimating the dimension of a model. *The Annals of Statistics*, 6 (2):461–464, 1978.

J. Sethuraman. A constructive definition of Dirichlet priors. *Statistica Sinica*, 4(2):639–650, 1994.

M. Stephens. *Bayesian Methods for Mixtures of Normal Distributions*. PhD thesis, University of Oxford, Oxford, 1997. Unpublished.

M. Stephens. Bayesian analysis of mixture models with an unknown number of components - an alternative to reversible jump methods. *The Annals of Statistics*, 28(1):40–74, 2000a.

M. Stephens. Dealing with label switching in mixture models. *Journal of the Royal Statistical Society, Series B*, 62(4):795–809, 2000b.

H. Thomas and J. J. Horton. Competency criteria and the class inclusion task: Modeling judgments and justifications. *Developmental Psychology*, 33 (6):1060–1073, 1997.

D. M. Titterington, A. F. M. Smith, and U. E. Makov. *Statistical Analysis of Finite Mixture Distributions*. Wiley, New York, 1985.

M. Wedel and W. S. DeSarbo. A review of recent developments in latent class regression models. In R. Bagozzi, editor, *Advanced Methods of Marketing Research*, pages 352–388. London: Blackwell Publishing, 1994.

C. F. J. Wu. On the convergence properites of the EM algorithm. *The Annals of Statistics*, 11(1):95–103, 1983.

D. S. Young, T. Benaglia, D. Chauveau, D. R. Hunter, R. T. Elmore, F. Xuan, T. P. Hettmansperger, and H. Thomas. *The* `mixtools` *Package: Tools for Mixture Models*, 2008. R Package Version 0.3.0.